News Popularity Beyond the Click-Through-Rate for Personalized Recommendations

Ashutosh Nayak
ashutosh.n1@samsung.com
Samsung R&D Institute India
Bengaluru, India

Mayur Garg
mayur.garg1@samsung.com
Samsung R&D Institute India
Bengaluru, India

Rajasekhara Reddy Duvvuru Muni
raja.duvvuru@samsung.com
Samsung R&D Institute India
Bengaluru, India

ABSTRACT

Popularity detection of news articles is critical for making relevant recommendations for users and drive user engagement for maximum business value. Among several well-known metrics such as likes, shares, comments, Click-Through-Rate (CTR) has evolved as a default metric of popularity. However, CTR is highly influenced by the probability of news articles getting an impression, which in turn depends on the recommendation algorithm. Furthermore, it does not consider the age of the news articles, which are highly perishable and also misses out on human contextual behavioral preferences towards news. Here, we use the MIND dataset, open sourced by Microsoft to investigate the existing metrics of popularity and propose six new metrics. Our aim is to create awareness about the different perspectives of measuring popularity while discussing the advantages and disadvantages of the proposed metrics with respect to the human click behavior. We evaluated the predictability of the proposed metrics in comparison to CTR prediction. We further evaluated the utility of the proposed metrics through different test cases. Our results indicate that by using appropriate popularity metrics, we can reduce the initial news corpus (item set) by 50% and still could achieve 99% of the total clicks as compared to unfiltered news corpus based recommender systems. Similarly, our results show that we can reduce the effective number of articles recommended per impression that could improve user experience with the news platforms. The metrics proposed in this paper can be useful in other contexts, especially in recommenders with perishable items e.g. video reels or blogs.

CCS CONCEPTS

• **Search Recommendation and Content Analysis** → News Recommender System; • **Humans and Interfaces** → Human Click Behavior; • **Machine Learning** → Popularity Detection.

KEYWORDS

News Lifecycle, News Popularity, Recommender Systems, User Click Behavior

SIGIR '23, July 23–27, 2023, Taipei, Taiwan

ACM ISBN 978-1-4503-9408-6/23/07…$15.00
https://doi.org/10.1145/3539618.3591741

ACM Reference Format:
Ashutosh Nayak, Mayur Garg, and Rajasekhara Reddy Duvvuru Muni. 2023. News Popularity Beyond the Click-Through-Rate for Personalized Recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '23), July 23–27, 2023, Taipei, Taiwan.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3539618.3591741

1 INTRODUCTION

Massive number of news articles are generated and posted online every day by news publishers, making it difficult for consumers to find interested news quickly [25]. Online news platforms such as Google News, Microsoft news and Apple news use News Recommender Systems (NRS) to provide curated news article to consumers by providing them with personalized news recommendations and popular news content. Click-Through-Rate (CTR) and the number of clicks are two well-known metrics for popularity detection. These metrics, though useful, have inherent bias and disadvantages. In this paper, we analyze existing metrics from the perspective of NRS, propose new metrics for defining popularity and discuss how these metrics can be used in NRS.

Currently, CTR is the default metric for measuring popularity of items in Recommender Systems (RS). CTR is defined as the ratio of number of clicks to the number of impressions. An impression is defined as an instance when a news article was recommended to a user. When a user clicks on a recommended article, it is defined as a click. Note that a click is not a guarantee that the user has read the article, however, we assume that a click shows user interest in the article. Recently, Perez et al. [26] proposed impressions as a measure of popularity and showed the utility of impressions as a metric in RS. Ability to measure the popularity of items (news articles) in a RS can be useful in recommender systems. For example, popularity is used in user cold start problems which makes it critical to measure popularity correctly since customer acquisition is one of the major expense for a firm [22]. Popularity detection can also be used to filter items (news articles) from product inventory (news publishers) to cater to the user base. Popularity detection can also be used in improving the recommendations in RS. It can help in reducing the number of recommendations to users as news less likely to be clicked will not be recommended during an impression. Popular items can also provide improved negative samples during training of RS as samples with higher likelihood of getting a click will be used [34].

To capitalize on the advantages of popularity detection, firms need to correctly identify the popularity of items in a RS. We focus on NRS and use items and news articles interchangeably. CTRs do not provide information as when the articles are clicked after their publication. Table 1 shows the statistics of the age of the news articles (from the time of publishing) at the time of impressions (no clicks) and clicks for different news categories. Statistically significant difference between the two samples indicate high probability that the distribution are different (*** refers to $p < 0.001$). This confirms that average age at the time of clicks is lower than the average age at the time of recommendations. Furthermore, it indicates that age of the article is an important consideration when designing the recommendation system. Results in Table 1 also highlight the heterogeneity in news articles across news categories. For example, articles on sports and video are highly perishable, that is, user lose interest with sports and video with time. However, users are still interested in articles on lifestyle for some time after publication.

Table 1: Age of the articles at impressions and clicks

Categories	Age @ Clicks (mean, std)	Age @ Impressions (mean, std)	T-statistic
Sports	13.08 (8.19)	14.32 (10.96)	36 ***
News	15.81 (15.12)	15.47 (11.36)	7.5 ***
Finance	18.47 (18.11)	23.30 (23.11)	30 ***
Travel	24.49 (21.65)	28.41 (22.85)	15 ***
Lifestyle	24.12 (25.27)	30.98 (30.47)	18 ***
Video	13.91 (13.27)	15.83 (18.13)	6.7 ***
T-statistic (Category wise)	19.41 (19)	23.76 (23.19)	89 ***

We are motivated by the results of Table 1 to include the age of the articles as an informative parameter for popularity detection. In this paper, we aim to define popularity of news articles from different perspectives. Towards this goal, this work builds different metrics for measuring popularity and discusses their utility, models for estimation and biases. Our work is more important in sectors when the number of items is large and new items are added regularly, for example, popularity of songs in music recommender systems (Spotify), popularity of reels in Short video reels (TikTok, YouTube), blog posts (LinkedIn) or product recommendation in ecommerce apps with unique products added by users (Etsy). This work presents the first evaluation of time-based metrics in measuring the popularity of news articles. The contribution of this paper are:

- We investigate existing popularity metrics for news articles and present their respective advantages and disadvantages
- We propose new metrics for measuring the popularity of news articles.
- We construct machine learning models to predict the proposed metrics
- We use different experiments to verify that using popularity metrics can help in improving news recommendations

Rest of the paper is organized as: we discuss the extant literature in related work on popularity metrics in Section 2. We provide the detailed methodology of this study along with proposed popularity metrics in Section 3. Next, we present the evaluation method for the utility of the proposed metrics in Section 4. We show results from the use of popularity metrics in recommender systems in Section 5. Finally, we conclude the paper with managerial insights, limitations of this work and future research in Section 6.

2 RELATED WORK

2.1 News Recommender System

Large number of news articles are published across the world by different publishers e.g. Washington Post, BBC news or New York Times. News portal e.g. Google News or Apple News aim to match the interest of their users (learnt by click behavior of the user) to the news articles (learn by the content of the news) and recommend relevant news articles to the users [39]. These recommendations by the news portals are made by NRS which run on powerful machine learning algorithms. There exists an extensive work in literature on improving the performance of the news RS including Multi-Interest Matching Network, or MINER [19], Multi-head Network with Self-Attention, or NRMS [40] or with Pre-trained models [42]. Liu et al. [21] use bayesian framework to match user iterest using clickstream data. [29, 41] presented a literature survey on the news RS. Microsoft open sourced a dataset from Microsoft News Website, called MIND, with 6 weeks data that includes consumer click behavior and news content [43]. We use the MIND dataset in our study for analysis and evaluation.

Recommender systems is a well studied field in literature and the field of improving recommender systems is an active area of research [45]. However, improvisation of news RS is not the focus of this research and building a RS is beyond the scope of this work. We focus on defining and measuring the popularity of the news articles which we believe could open new directions in building recommender systems. We discuss the extant literature on popularity metrics next.

2.2 Popularity Metrics

Measuring popularity is an important area of research in recommender systems. Firms need different metrics for measuring the popularitys. Saini et al. [31] studied the relation between different popularity metrics in softwares. They studied popularity of github repositories and identifed that not all metrics point to same items as popular. For example, number of downloads for a repository is not associated with development activity. Similarly, Chatzopoulou et al. [2] discovered that average rating of a YouTube video is not correlated with other metrics e.g. number of comments or favorites. Ability to measure the popularity of items in a recommender system is a useful tool. The list of the systems that use popularity is very exhaustive so we list some examples here. The most common business use case of item popularity is user cold start problem where new users are recommended popular items [10]. With an increase in the usage of social media, firms are using popularity to assess the virality of posts [5]. Li et al. [20] discuss how the popularity is used to recommend coupons to coupon vendors like web pages. Popularity is also used in music and podcast recommendations [17].

Click-Through-Rate (CTR) is the most common measure for popularity and CTR prediction is an active area of research [32, 46]. CTR has been used in wide range of applications e.g. sponsored search [35] or display advertisements [3]. Konig et al [16] build a regression tree model to predict the CTR. However, CTR suffers from impression bias as a product can get a CTR>0 only if it was recommended to users (details in Section 3.2. Other common measures of popularity include the number of clicks or purchases for an item, importance of node in a network [6, 18], user engagement with items e.g. shares and comments [11] or quality of the item (e.g. length of the video watched [44]).

Depending on the dataset, different platforms have different measures of popularity. For example, ratings in terms of number of stars (e.g. products in Amazon [12]), number of likes (videos on YouTube [33]), top most played podcasts (e.g. podcasts in Spotify [23]) or the number of shares or comments (posts in Facebook or tweets in Twitter [13]). However, these measures of popularity are not useful in news platforms because of the nature of the news articles and the platform itself. News platforms do not have features like social media (likes, shares, or user network), although, they can provide more personalization unlike offline media (newspapers). Next, we discuss popularity specifically in the context of NRS.

2.3 Popularity in News Recommender Systems

Understanding the popularity of news articles is important not only for the news platforms but also for the news publishers. Study on how news articles popularity can be defined, what its characteristics are, and whether it can be predicted is at its nascent stage. This is mainly due to lack of open source dataset with relevant information in news recommenders. To measure the popularity of news articles in NRS, we need the information on the characteristics of news articles, users and context. While the existing datasets have information on the news articles (content) and users (click behavior), we lacked information on the context (news recommended but not clicked, time of the impressions, or age of articles in an impression) as pointed by Perez et al. [26]. In this work, we aim to provide different perspectives on measuring popularity for news articles. We also discuss how to use popularity metrics in NRS.

News popularity has been defined in various ways in the extant literature. Tsangkias et al [36] used number of comments in a news article as a measure of popularity. Similarly, Tsangkias et al [37] used the volume of comments for popularity. Naseri et al [24] used the number of views in Telegram channel for popularity. Keneshloo et al. [15] also used number of views in day one after publication as a measure of popularity. They use temporal features from clickstream data e.g. views in first 30 minutes after the publication. Bhatia [1] used number of page views, tweet counts and likes of article links for measuring their popularity. Ghosh et al [8] use page load requests as popularity metric. Survivability, or lifespan of news article is also as a popularity metric for news articles [38]. They use data from a social media platform e.g. likes, comments and shares to predict the popularity. Perez et al [26] used the number of impressions as a measure of popularity in their study to showcase the utility of news popularity. An impression is similar to a view, however, an impressions is limited by the recommendation by the RS. Note that news platforms do not have features available in social media e.g. likes, shares, comments and user connectivity.

Existing works also consider popularity in building their NRS. Qi et al. [28] proposed PPREC, a news recommender based on popularity score of a news article in neural network architecture. They considered news content, recency and CTR for popularity score. However, they do not explicitly estimate for popularity score but use it as an argument to find similarity scores between user and recommendations. As the authors note, CTR will suffer for newly published articles (which all have 0 impressions and hence, 0 CTR). Similarly, Cho et al. [4] considered poularity measures such as freshness and CTR in NRS and observe improvement in performance when augmented with popularity metrics. Building on these works, we discuss the role of popularity metrics in NRS and consider metrics beyond the well-known CTR while considering human click bahavior.

Next, we present the methodology used in this paper to introduce and evaluate new metrics for measuring popularity.

3 METHODOLOGY

We use the MIND Large dataset in our study. The popularity metrics in this study required the information on impressions (timestamp when a set of articles were recommended to a user) and content of the news. MIND dataset allows us to investigate different popularity metrics and propose new metrics. Detailed explanation of the dataset is presented in Wu et al. [40]. All the experiments and models in this study were built on Windows 11 platform, i7 Intel processor with 16 GB RAM. We use Python 3.6 for all the analysis. We use pytorch 1.13 for building neural network models.

3.1 Dataset

Authors of the MIND dataset classify the news articles into 17 different categories. They further classify a news categories into respective subcategories. For reducing the complexity of our analysis, we pick top six news article categories (based on the count of articles in the training dataset) that account for 80% of the total articles and 90% of total impressions. In the rest of the paper, we show results for these six news categories – Sports, News, Finance, Travel, Lifestyle and Video. Since we do not have click/no click information in the test dataset, we use MINDtrain and MINDdev data provided in the MIND dataset for our analysis. As the dataset contains information for 7 days, there is data truncation in calculating the lifespan of news articles (news published on the sixth day in the dev dataset can have a maximum lifespan of 24 hours). To overcome the truncation issue, we consider the news articles that had at least one impression on the first day of the dataset. Similarly, to reduce the computational complexity, we only consider users who had at least one impression on the first day in dataset. Our data contains information from 140000 users with 2.2 million impressions. These users account for approximately 20% of the total users and 30% of total impressions in the data set.

The dataset contains information on news content (news title, abstract, news category and news subcategory and Translating Embeddings (TransE embeddings) of the entities in the news title and abstract. User information includes click behavior (history of the news articles clicked in the first four months during the period of data collection). Data set does not include any demographics (age, gender, location) data for the user. The dataset presents contextual information in the form of impressions. An impression is an instance when a user checks the Microsoft news platform. The algorithm used by Microsoft recommends news articles to the user, based on their click behavior and content of the news articles in the repository. An impression also contains UNIX timestamp information. We use this information to extract the exact time of the impression in the user journey through news portal.

We use the available information to impute the age of the news articles at the time of an impression. Since the time of publication of an article is not available in the dataset, we use the first impression of an article as the time of publication. We calculate the age of a news article at the time of the impression as the difference between the time of the impression and the time of the publication of that news article. Similar to first impression, we use the last impression of an article as its death (news article is no more of interest to the user base). Thus, we use the first and the last impression of a news article to calculate the lifespan. Using the impression timestamp, we also obtain the time of the day and the day of the week from the impression timestamp.

3.2 Model-Free Analysis

In this Section, we provide a model-free evidence of the heterogeneity in news articles across categories and how the click behavior of a news article change with time. Before discussing time dependency, we first show statistics for most widely used metric – Click-Through-Rate (CTR). Figure 1 shows the lifetime CTR (ratio of total number of clicks to the total number of impressions) for different news categories.

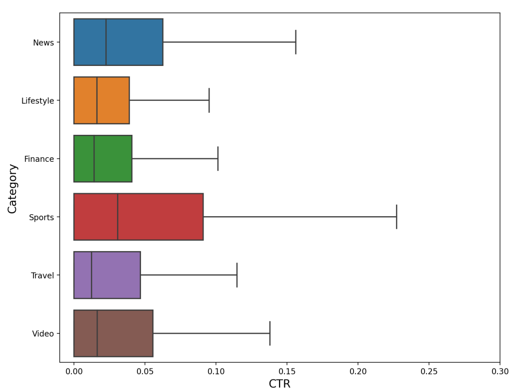
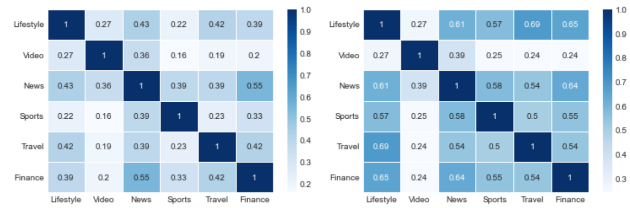


Figure 1: CTR for Articles in Different Categories

However, there are two major issues with using CTR as a popularity metric: (1) CTR is dependent on impressions, that is, CTR of a news article is 0 if it has zero impressions. Number of impressions is a manifestation of the algorithm used by RS [26]. For example, if algorithm 1 never recommends a news article to its users while algorithm 2 does, news article will have a chance of getting a click (or CTR>0) only with algorithm 2. (2) CTR is not necessarily correlated with the number of impressions. In the dataset we consider, we observe that the correlation between CTR and number of impressions is 0.005. We check a linear regression model $CTR_n = \alpha + \beta NumberImpressions_n$ and observe that $\beta = 0.005$ is statistically insignificant, thus establishing no linear relationship between CTR and Impressions. This shows that while having an impression is a prerequisite for a click, it does not guarantee clicks for the articles.



(a) User-wise **(b) Impression-wise**

Figure 2: Correlation Between Categories Based on Impressions and User Behavior

As CTR depends on impressions, impressions are critical in matching the interest of the user. The correlation structure between the news categories based on impressions (articles recommended by the base recommender) and user past click behavior (articles clicked by users in the past) is shown in Figure 2. The difference in the magnitude of the correlation numbers in Figure 2 illustrate that the base recommender recommends a wider range of news categories to the users, however, users have focussed interest in certain news categories. This indicates discrepancy in the news recommended vs the user interests.

Number of clicks is also a widely used metric for measuring the popularity as most clicked items are deemed popular as an indication that popular items are able to attract interests from multiple users. First issue of CTR is also observed in using the number of clicks as a measure of popularity because an article can have clicks only when it has impressions. However, the number of clicks is correlaetd with the number of impressions as we observe a correlation of 0.72 in the data set. In Figure 3, we show the average number of clicks and impressions per news articles for different categories. Similar to CTR in Figure 1, these metrics provide aggregate information but do not provide time-based information.

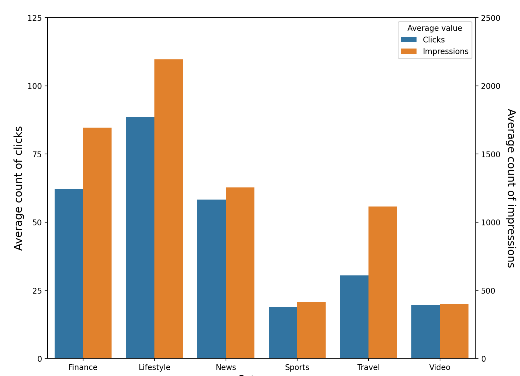


Figure 3: Average Number of Impressions and Clicks

To understand the lifecycle of news articles in terms of number of clicks and impressions, we show how these metrics vary over time in Figure 4. It shows heterogenerity across categories. While CTR of articles in Figure 4 generally decrease with age of the article, CTR for "Lifestyle" category does not follow this trend. It also shows that articles take time to reach the peak of number of clicks or impressions (we discuss this further in Section 3.3.3). It also shows that number of clicks for an articles follows similar trend as the number of impressions.

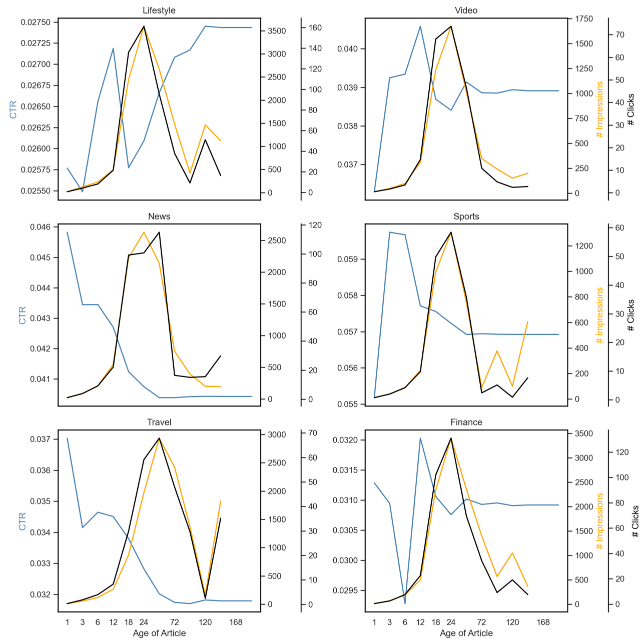


Figure 4: Average CTR, Number of Impressions and Clicks by the Age of the News Articles (best viewed in color)

3.3 Proposed Metrics

In this Section, we propose different metrics for defining and measuring the popularity of news articles. We also discuss the advantages and biases for each of the metrics. One common issue with all the metrics is that we are using impressions data, provided by the existing algorithm (which is not known to researchers). Thus all the metrics have indirect and direct effect of the manifestation of the algorithm used in generating the MIND dataset.

3.3.1 Lifespan of News Articles. Lifespan of a news article is defined as the length of the time for which a news articles remains relevant (keeps users interested) in NRS. Note that we do not know from the data set when the article was published or removed from the news inventory. However, we consider the first impression ($Impression_n^{first}$) and the last impression ($Impression_n^{last}$) of a news article n to impute its lifespan (Equation 1).

$$lifespan_n = Impression_n^{last} - Impression_n^{first} \quad (1)$$

The distribution of the lifespan of the news articles for different categories is shown in Figure 5. As observed in Figure 5, NRS recommends certain news articles for longer period (e.g. lifestyle) as compared to other news articles (e.g. sports). Lifepan can be used in RS as items with higher lifespan could be considered for recommendation for longer time. This claim is also supported by Figure 3 which shows that news categories with longer lifespan on average has higher number of clicks.

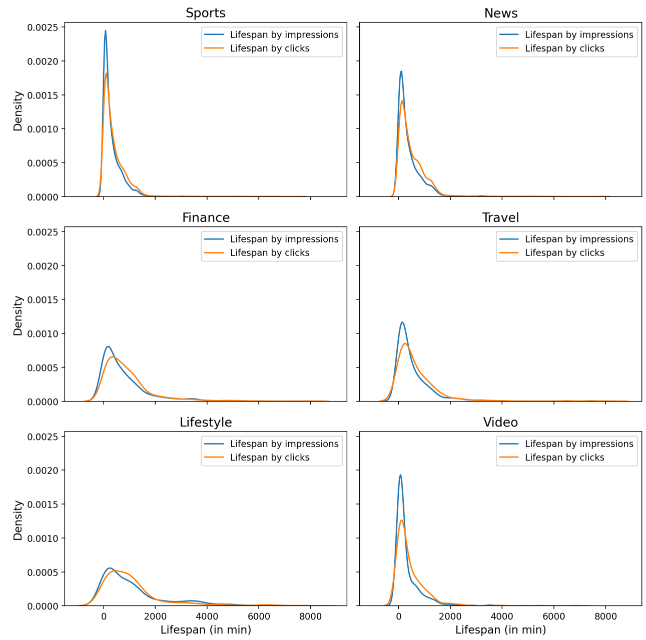


Figure 5: Distribution of Impressions and Clicks Lifespan of News Articles for Different Categories

However, lifespan again depends on the recommendation engine (algorithms used in RS). This also makes it very difficult to predict [38]. Subsequently, while the lifespan can express how long a news articles can be considered for recommendation, it does not provide any temporal dynamics about the user interests of the news articles. To include temporal dynamics, we next present the half-life for news articles.

3.3.2 Time to Achieve Half-Life for Clicks/Impressions. We define impressions (or clicks) Half-Life of a news article as the time taken by a news article to reach half of its total impressions (or clicks). Half-Life is expressed mathematically in Equation 2 as $t_{half,n}$ where $Impression_{n,t}$ is the number of impressions t hours after publishing and $T = Impression_n^{last}$. Similarly, it could be defined for the number of clicks.

$$t_n^{half} = max\{t'\} \ni \sum_{t=0}^{t=t'} Impression_{n,t} \leq \frac{1}{2} \sum_{t=t'}^{t=T} Impression_{n,t} \quad (2)$$

Idea behind the Half-life metric is same as in nuclear physics where half-life is used to identify the rate of radioactive decay in unstable atoms [9]. Similarly, Half-life of news articles can be used to identify how fast users lose interest in news articles after being published. The distribution of Half-life for news articles in terms of impressions and clicks is shown in Figure 6.

Half-life metric provides information on the rate at which users may lose interest in a news article. However, as observed in Figure 4, half-life information misses out in deciphering the time at which user interest may peak. Next, we discuss another metric to identify the peak time for a news article.

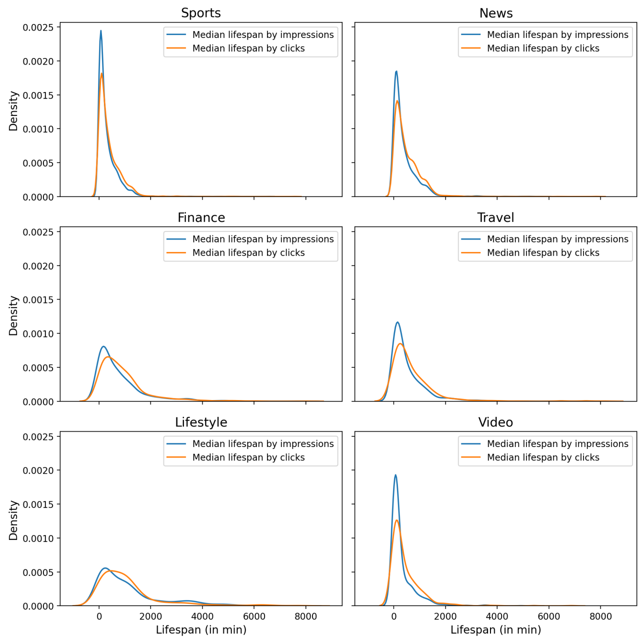


Figure 6: Distribution of Half-Life of Impressions and Clicks Count for News Articles in Different Categories

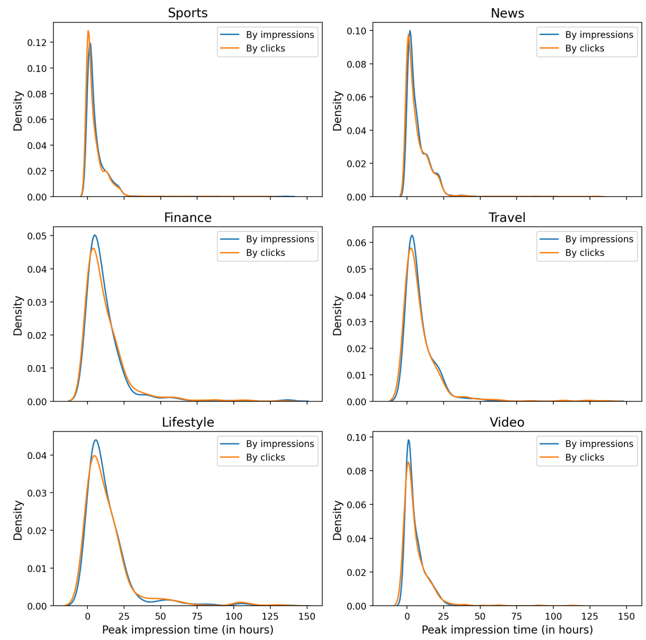


Figure 7: Distribution of Peak-Time of Impressions and Clicks for Different News Categories

3.3.3 Time to Achieve Peak-Time for Clicks/Impressions. We define Peak-Time for impressions (or clicks) as the time taken by a news articles after publishing to achieve its peak number of impressions (or clicks). We present Peak-Time in Equation 3 where $Impress_{n,t} = 1$ if article n was recommended after t hours from publishing. We use moving average in our experiments to calculate Peak-Time.

$$
\begin{aligned}
t_n^{peak} &= t' \ni \sum_{n=1}^{n=N} Impress_{n,t} \\
&= max\left(\sum_{n=1}^{n=N} Impress_{n,t} \forall t \in \{1, 2, ..., T\}\right)
\end{aligned} \tag{3}
$$

As seen in Figure 4, the number of clicks follow similar pattern as the number of impressions. We find similar insights from the the distribution of the time to reach peak number of impressions and clicks for different news categories. We present the Peak-Time distribution for impressions in Figure 7.

Time to achieve Peak number of impressions (clicks) indicates the maturity time of news articles, that is, when articles achieve maximum user interests. As expected from the previous analysis (Section 3.3.2), news and sports reach their Peak-Time earlier than other news categories. Peak-Time could suggest when a certain news article could be considered for recommendation (more weightage during Peak-Time). However, as we discuss in results, Peak-Time metric is very difficult to estimate for newly published articles. This could be attributed to the dependency of the Peak-Time on the clicks of the articles.

All the previous metrics proposed in this paper do not consider user information. Next, we discuss a coverage metric which considers past user click behavior as user information.

3.3.4 User Coverage. Coverage is an important and well-studied metric for recommender systems [7]. Coverage in RS is defined as the diversity of the items recommended to the users (e.g. different types of items). Coverage is defined from a user's perspective, hence it is the coverage of items. We define a new metric from the perspective of the items (articles in NRS) as the coverage of users. Thus, we define user coverage as the different types of users that might be interested in a news article. In other words, we define user coverage as the number of types of users to whom a news article would be appealing.

We build an auto-encoder model with encoder-decoder framework to segment the users into different types. We consider the number of clicks in the six news categories, $Behavior_u = \{Behavior_{u,c}, \forall c \in \{1, 2, ..., 6\}\}$, to represent the click behavior of user u. We use latent vectors, $Latent_u$, to represent a user's interest (dimensionality reduction technique to represent users). The encoder and decoder function used to learn the latent representation of a user's interest in news articles is shown in Equation 4 and Equation 5 where U is the set of users. We use neural networks to model the encoder function (*Encoder*) and the decoder function (*Decoder*).

$$Latent_u = Encoder(Behavior_u) \ \forall u \in U \tag{4}$$

$$Behavior_u = Decoder(Latent_u) \ \forall u \in U \tag{5}$$

For visualization, we use a 2-dimensional latent space as shown in Figure 8 to represent the nature of a user (or user interest). We use K-means algorithm to group users into 10 segments. Different colors in Figure 8 shows different user segments. We show example of users interested in certain article categories in Figure 9. For example, we consider that a user is interested in sports if the number of articles clicked in sports categories is more than 75% of all the clicked articles by that user. We consider a threshold, th, such that if a user segment g has a CTR of greater than th for a news article n, we consider that users in g are interested n.

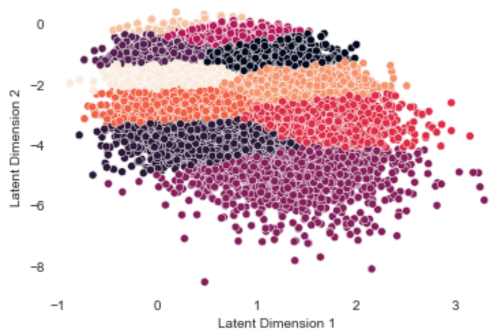


Figure 8: Representing Users in a 2-Dimensional Latent Space (Best seen in Color).

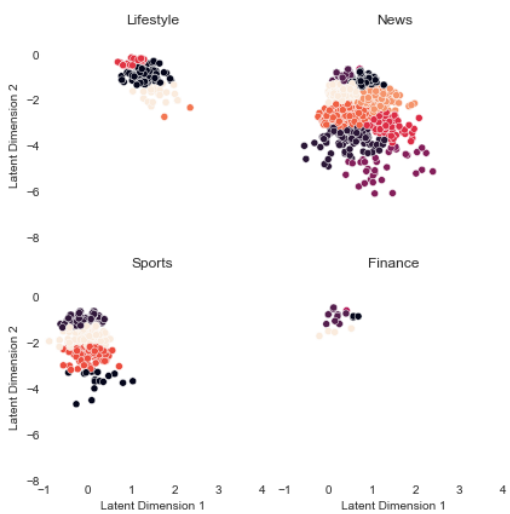


Figure 9: Example of Users Interested in Lifestyle, News, Sports and Finance (Best seen in Color).

User coverage is novel metric that identifies user groups who will find an item interesting, thus algorithms should consider these articles during personalized recommendation. Along with recommender systems, similar metrics could be used in other problems that require user segmentation (e.g. advertisement, churn prediction). We discuss one use case of user coverage for cold start problems in Section 4.1.3.

3.3.5 Normalized Clicks. We define $NormalizedClicks$ for a news article n as the ratio of the number of clicks of n to the lifespan of n. NC is shown in Equation 6. We combine a well-known metric (number of clicks) with lifespan of news articles to propose a new metric. This metric can be calculated for filtering news to bring news articles into the recommendation engine (more details in Section 4.1.1).

$$NormalizedClicks_n = \frac{NumberClicks_n}{Lifespan_n}$$

Higher values of NC indicate high popularity for the respective news articles. NC normalizes over lifespan to account for perishability of news articles. However, NC has a disadvantage that it penalizes higher lifespan of news which may be higher because the users are still interested in the article.

3.3.6 Reading Time for News Articles. In many RS, we use the click behavior of users and assume users clicked on the items they were interested in. However, a click does not guarantee user interest. We postulate that time taken by users to read a news article could be an informative metric for measuring the news popularity. Due to data limitations in calculating reading time of news articles in MIND, we do not dicuss this further and leave it here as future research direction.

4 EVALUATING PROPOSED METRICS

In this Section, we discuss how can we use the popularity metrics proposed in this study to improve News Recommender Systems (NRS). We present three case studies on the use of popularity metrics. Note that it is beyond the scope of this work to build a RS based on popularity detection but we discuss the potential directions as how popularity can help in improving performance of the RS. However, to use these popularity metrics, we need to estimate the metrics. We do so by building different machine learning models in Section 4.2.

4.1 Case Studies

In this work, our aim is create awareness about different perspectives in detecting popularity. We first discuss three case studies where proposed metrics could be used directly. Similar to Perez et al [26], we use re-ranking of recomendations list from the base recommender (recommender used in MIND dataset) to test the performance of the popularity metrics proposed in this paper.

4.1.1 Filtering News Articles. News Recommender Systems (NRS) collect and purchase news from different publishers and store them in their news corpus for recommendations, model development and model update. We can use popularity metrics to identify which news articles will not be appealing to the user base and hence should not be included in corpus. We measure the performance of filtering by calculating the percentage of news articles removed by removing least popular articles.

4.1.2 Using Popularity in Recommendations. RS help users in finding interested news articles from myriad of news articles. To capture the user interest, news platforms recommend a list of news articles and users click on the interested news. The purpose of RS will be lost if the user has to select items from a long list of recommendations. We postulate that recommending relevant news articles in a shorter list could improve consumer experience [27]. We test the performance of the proposed metrics by checking if we can reduce the size of recommendation list in an impression by filtering out non-popular news articles (based on the age of the articles).

4.1.3 User Coverage for Cold Start Problem. User cold starts problem arises when the system does not have enough information about the users. The most commonly used method is recommending popular items. We postulate that since we do not have enough information about the new users, recommending news article that cover more user segments could increase the chance of finding interested news for these users. We also check if the base recommender is recommending news articles to user segments that might not be interested in that news. We test the performance of the popularity metrics proposed in this paper to check if the clicked articles in impressions are also the popular items.

4.2 Model Development

We use the content of the news articles as exogenous variables to model different popularity metrics. We use the first impression of a news article to obtain its launch time (time of the day and day of the week). Our categorical variables include: long abstract (= 1 if the abstract has more than 500 words) and one hot encoding for news categories and subcategories. We use time of launch and day of launch in launch time variable. We use # to represent count. We represent news articles by sentence embeddings including Sentence-BERT embeddings [30] and fasttext embeddings [14]. Data summary for the popularity metrics is shown in Table 2.

Table 2: Data Summary for Popularity Metrics

Variable	Mean	Median	Max	std
CTR	0.048	0.027	0.66	0.06
Lifespan (Minutes)	1266	644	8639	1739
Half-Life (Minutes)	456	223	7856	659
Peak-Time (Hours)	6.13	2	138	10.51
# Clicks	48.47	2	10630	318.2
# Clicks/Lifespan	1.15	0.12	225.4	6.42
# Impressions	238.6	6	23037	1041.2

We use different machine learning algorithms to model the popularity metrics including Random Forest, Ridge regression, Neural Networks, K-Nearest Neighbours and Generalized Linear Models. We discuss the performance of these algorithms and results from the case studies in the next Section. Note that due to space constraints, we report the results from the best performing model.

5 RESULTS AND DISCUSSIONS

5.1 Model Performance

We summarize the performance from different models in Table 3. We use different predictor variables and we add them to the list of variables as presented in Table 3. We only report the R-square values for the best performing model from the different model. As CTR is the most well-known metric, we use results for CTR prediction as a baseline to compare the performance of different models. We are the first in studying the aforementioned metrics over the context of NRS which could be used as baseline in future works. We **bold** the results for models which perform better than the CTR.

Table 3: Model Performance for Popularity Metrics

	Metrics				
Variables	CTR	Lifespan	Half-Life	Peak-Time	#Clicks/ Lifespan
Embeddings	0.069	0.037	0.017	0.033	0.036
+Abstract	0.075	0.058	0.031	0.04	0.029
+Category	0.076	0.073	0.046	0.051	0.033
+Subcategory	0.081	**0.221**	**0.129**	0.072	0.034
+Launch Time	0.084	**0.326**	**0.158**	**0.097**	0.035
+Title	0.082	**0.222**	**0.132**	0.073	0.034
+# Entity	0.085	**0.330**	**0.162**	**0.10**	0.035

Results in Table 3 show how the algorithms perform when we use different set of predictor variables. R-Square values also indicate that these metrics are difficult to estimate and we believe state-of-the-art Natural Language Processing (NLP) can significantly improve the performance of the models. Most of the existing works frame the CTR prediction problem as classification problem. For each of the metrics in Table 3, we convert the response variables into dichotomous variables (equal to 1 if the response variable is more than approximately 50th percentile of the respective response variable in the training data set). The results for the classification problem is shown in Table 4. We use all the variables from Table 3 for models used in classification.

Table 4: Modeling Metric as Classification Problem

Metrics	Accuracy	F1-Score	AUC
CTR	0.61	0.66	0.62
Lifespan	**0.62**	0.49	**0.66**
Half-Life	0.58	0.53	**0.64**
Peak-Time	0.59	0.50	**0.64**
# Clicks/Lifespan	0.59	0.57	0.62

5.2 Using Popularity Metrics

Popularity metrics can be a useful tool in RS. In this Section, we show how metrics can be used. We present the results from the models constructed in Section 4.2. Table 5 shows the % decrease in the number of clicks as compared to the baseline recommender when we remove least popular $bottom\%$articles from the corpus. Our results indicate that using appropriate popularity metrics can help in reducing news corpus. For example, if we remove least popular 60% of articles, we can still achieve almost 99% of total clicks. News recommenders can also purchase better corpus and effective reduction in total clicks could be reduced further. Normalized Clicks metric appears to perform better than CTR as the reduction in the number of clicks is lower than that of using CTR.

Relevant articles in a smaller list of recommendations can improve user experience. We consider time based popularity metrics to achieve this capability in RS. In Table 6, we consider the reduction in number of clicks as compared to base recommenders when the age of the articles is h hours after the respective metric. Here,

Table 5: Results for Reducing the News Corpus

	$bottom = 40\%$ (True, Model)	$bottom = 50\%$ (True, Model)	$bottom = 60\%$ (True, Model)
CTR	(4.4%,32.6%)	(16.3%,41.7%)	(31.5%,54%)
Lifespan	(**3.4%**,40.1%)	(**5.9%**,45.4%)	(**11.3%,50.9%**)
Half-Life	(**3.2%**,42.5%)	(**6.1%**,48.6%)	(**11.6%**,55.5%)
Peak-Time	(**3.2%**,36.1%)	(**4.7%**,50.7%)	(**12.1%**,56.6%)
#Click/Lifespan	(**0.26%,7.3%**)	(**0.74%,17.1%**)	(**1.5%,45.3%**)
#Impressions	(**0.5%,30.3%**)	(**0.9%**,42.9%)	(**1.4%,51.2%**)

we use the true value for metrics from the data set as we did not construct models for Time-based CTR. For Time-Based CTR, we check the % decrease in the number of clicks once the CTR falls below c% from the category average CTR. For Lifespan, we check the % decrease in the number of clicks once the age of the article crosses $(1 - c)$% from the category average lifepan. Results show that the % drop in the number of impressions is higher than the % drop in the number of clicks. Also, Time-Based CTR performs better as the ratio of % decrease in clicks is lower than that of using Lifespan when compared to % drop in number of impressions.

Table 6: Results for Reducing the Number of Impressions

	(#Clicks, #Impressions)		
	$h = 3\,hour$	$h = 6\,hour$	$h = 12\,hour$
Half-Life	(21.7%,24.5%)	(8.5%,11.1%)	(3.5%,4.9%)
Peak-Time	(43.4%,49.9%)	(16.9%,22.1%)	(7.1%,9.9%)
	$th = 40\%$	$th = 30\%$	$th = 20\%$
Lifespan	(54.9%,56.3%)	(45.8%,47.4%)	(34.8%,36.7%)
Time-based CTR	(37.1%,62.8%)	(26.4%,51.2%)	(15.6%,36.5%)

For user coverage, we consider that a news article is appealing to (or provide coverage to) a user segment if the CTR in that segment is greater than a threshold CTR_o. We observe that the average number of user segments covered by clicked articles is higher than the average number of user segments covered by news articles that were recommended but not clicked (7). Note that the number of users covered for clicked articles is stronger for cold start users. This finding is backed by the theoretical foundation that for a user u with limited information, recommeding articles that appeal to a wider range of users could increase the odds of u being interested in that article.

Table 7: Number of User Segments Covered

Users	$CTR_o = 0.03$		$CTR_o = 0.04$	
	Clicked	Impressions	Cliked	Impressions
All	7.16	4.43	6.25	3.33
Cold Start	7.51	4.54	6.66	3.44

In the final result, we check for cold start users if we remove articles that cover less than *segment* user segments, what is the % drop in number of clicks as compared to the number of impressions. We $CTR_o = 0.04$. Results in Table 8 show that the % drop for impressions is higher than that of clicks, indicating that user clicks can be achieved in smaller recommendations list in an impression.

Table 8: Results for Reducing Number of Impressions for Cold Start Users

	segment			
	3	4	5	6
% drop in #Clicks	21.1%	26.5%	30.9%	36.9%
% drop in #Impressions	52.3%	59.9%	65.1%	71.1%

6 CONCLUSION

Popularity of items (news articles) is a useful tool for recommender systems (News Recommender Systems, NRS). News articles differ from other product-based recommender systems because of its perishability and heterogeneity across news categories. Users usually show low interest in older news for some categories but articles in some categories appeal to users for longer periods. In this work, we focus on defining different metrics to measure the popularity of news articles. We use open source MIND dataset released by Microsoft in this work.

In this work, we provide an exhaustive descriptive analysis on different popularity metrics based on different perspectives. Following this analysis, we present proposed metrics for popularity and discuss potential advantages and disadvantages of these metrics. We discuss three use cases of measuring the popularity of news articles. First, it can help the news platforms to filter and collate only the popular news articles from different publishers around the world. This could help NRS in reducing the cost of purchasing articles from the publishers and negotiating purchase terms. Second, we show that we can use time-based popularity metrics to reduce the number of recommendations. It can increase user satisfaction if interested news articles are included in fewer recommendations. Second, it can help in improving recommendations for cold start users (or users we do not have enough information from click behavior) by showing popular news that cover more types of users. We believe the metrics defined for measuring the popularity are generalizable and can be extended to other recommender systems, particularly where millions of items are generated in short time, especially user generated content (e.g. video reels, art works, blogs).

This work has certain limitations, which also underlines potential future research directions. While the preliminary analysis shows that the popularity metrics is useful in news RS, this work does not focus on building an improved NRS that uses the proposed popularity metrics. We base our analysis on NRS using the MIND dataset, but a follow up work can verify the generalizability of the proposed popularity metrics in other product recommendation systems. Finally, we define popularity metrics based on the content of the news articles. However, popularity of articles may vary depending upon the user base. Hence, another direction is to develop new popularity metrics that includes behavior of its users.

REFERENCES

[1] Sejal Bhatia. 2021. Application and evaluation of Machine Learning for news article popularity prediction. In *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*. IEEE, 1–5.

[2] Gloria Chatzopoulou, Cheng Sheng, and Michalis Faloutsos. 2010. A First Step Towards Understanding Popularity in YouTube. In *2010 INFOCOM IEEE Conference on Computer Communications Workshops*. 1–6. https://doi.org/10.1109/INFCOMW.2010.5466701

[3] Junxuan Chen, Baigui Sun, Hao Li, Hongtao Lu, and Xian-Sheng Hua. 2016. Deep ctr prediction in display advertising. In *Proceedings of the 24th ACM international conference on Multimedia*. 811–820.

[4] Sungmin Cho, Hongjun Lim, Keunchan Park, Sungjoo Yoo, and Eunhyeok Park. 2021. On the Overlooked Significance of Underutilized Contextual Features in Recent News Recommendation Models. *arXiv preprint arXiv:2112.14370* (2021).

[5] Seyed Pouyan Eslami, Maryam Ghasemaghaei, and Khaled Hassanein. 2022. Understanding consumer engagement in social media: The role of product lifecycle. *Decision Support Systems* 162 (2022), 113707.

[6] Kristen Forbes. 2016. Examining the beauty industry's use of social influencers. *Elon Journal of Undergraduate Research in Communications* 7, 2 (2016), 78–87.

[7] Mouzhi Ge, Carla Delgado-Battenfeld, and Dietmar Jannach. 2010. Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*. 257–260.

[8] Sayar Ghosh Roy, Anshul Padhi, Risubh Jain, Manish Gupta, and Vasudeva Varma. 2022. Towards Proactively Forecasting Sentence-Specific Information Popularity within Online News Documents. In *Proceedings of the 33rd ACM Conference on Hypertext and Social Media*. 11–20.

[9] Harry Godwin. 1962. Half-life of radiocarbon. *Nature* 195 (1962), 984–984.

[10] Jyotirmoy Gope and Sanjay Kumar Jain. 2017. A survey on solving cold start problem in recommender systems. In *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 133–138.

[11] Mario Haim, Anna Sophie Kümpel, and Hans-Bernd Brosius. 2018. Popularity cues in online media: A review of conceptualizations, operationalizations, and general effects. *Studies in Communication| Media* 7, 2 (2018), 186–207.

[12] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*. 507–517.

[13] Maximilian Jenders, Gjergji Kasneci, and Felix Naumann. 2013. Analyzing and predicting viral tweets. In *Proceedings of the 22nd international conference on world wide web*. 657–664.

[14] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759* (2016).

[15] Yaser Keneshloo, Shuguang Wang, Eui-Hong Han, and Naren Ramakrishnan. 2016. Predicting the popularity of news articles. In *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 441–449.

[16] Arnd Christian König, Michael Gamon, and Qiang Wu. 2009. Click-through prediction for news queries. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. 347–354.

[17] Junghyuk Lee and Jong-Seok Lee. 2018. Music popularity: Metrics, characteristics, and audio-based prediction. *IEEE Transactions on Multimedia* 20, 11 (2018), 3173–3182.

[18] Kristina Lerman and Tad Hogg. 2010. Using a model of social dynamics to predict popularity of news. In *Proceedings of the 19th international conference on World wide web*. 621–630.

[19] Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. 2022. MINER: Multi-Interest Matching Network for News Recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*. 343–352.

[20] Yung-Ming Li, Jyh-Hwa Liou, and Ching-Yuan Ni. 2019. Diffusing mobile coupons with social endorsing mechanism. *Decision Support Systems* 117 (2019), 87–99.

[21] Jiahui Liu, Peter Dolan, and Elin Rønby Pedersen. 2010. Personalized news recommendation based on click behavior. In *Proceedings of the 15th international conference on Intelligent user interfaces*. 31–40.

[22] Erik Maier and Jaap Wieringa. 2021. Acquiring customers through online marketplaces? The effect of marketplace sales on sales in a retailer's own channels. *International Journal of Research in Marketing* 38, 2 (2021), 311–328.

[23] Huanru Henry Mao, Shuyang Li, Julian McAuley, and Garrison Cottrell. 2020. Speech recognition and multi-speaker diarization of long conversations. *arXiv preprint arXiv:2005.08072* (2020).

[24] Mohammad Naseri and Hamed Zamani. 2019. Analyzing and predicting news popularity in an instant messaging service. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1053–1056.

[25] Shumpei Okura, Yukihiro Tagami, Shingo Ono, and Akira Tajima. 2017. Embedding-based news recommendation for millions of users. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 1933–1942.

[26] Fernando Benjamin Perez Maurera, Maurizio Ferrari Dacrema, and Paolo Cremonesi. 2022. Towards the Evaluation of Recommender Systems with Impressions. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 610–615.

[27] Stan Phelps. [n. d.]. Take A Cue From In-N-Out And Chick-fil-A To Withhold And Differentiate In A Sea Of Sameness. *Forbes* ([n. d.]). https://www.forbes.com/sites/stanphelps/2018/04/18/take-a-cue-from-in-n-out-and-chick-fil-a-to-withhold-and-differentiate-in-a-sea-of-sameness/?sh=6d5591b11e14

[28] Tao Qi, Fangzhao Wu, Chuhan Wu, and Yongfeng Huang. 2021. Pp-rec: News recommendation with personalized user interest and time-aware news popularity. *arXiv preprint arXiv:2106.01300* (2021).

[29] Shaina Raza and Chen Ding. 2021. News recommender system: a review of recent progress, challenges, and opportunities. *Artificial Intelligence Review* (2021), 1–52.

[30] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[31] Munish Saini, Rohan Verma, Antarpuneet Singh, and Kuljit Kaur Chahal. 2020. Investigating diversity and impact of the popularity metrics for ranking software packages. *Journal of Software: Evolution and Process* 32, 9 (2020), e2265.

[32] Qingquan Song, Dehua Cheng, Hanning Zhou, Jiyan Yang, Yuandong Tian, and Xia Hu. 2020. Towards automated neural interaction discovery for click-through rate prediction. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 945–955.

[33] Dinuka A Soysa, Denis Guangyin Chen, Oscar C Au, and Amine Bermak. 2013. Predicting YouTube content popularity via Facebook data: A network spread model for optimizing multimedia delivery. In *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 214–221.

[34] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[35] Ilya Trofimov, Anna Kornetova, and Valery Topinskiy. 2012. Using boosted trees for click-through rate prediction for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy*. 1–6.

[36] Manos Tsagkias, Wouter Weerkamp, and Maarten De Rijke. 2009. Predicting the volume of comments on online news stories. In *Proceedings of the 18th ACM conference on Information and knowledge management*. 1765–1768.

[37] Manos Tsagkias, Wouter Weerkamp, and Maarten de Rijke. 2010. News comments: Exploring, modeling, and online prediction. In *European Conference on Information Retrieval*. Springer, 191–203.

[38] T Franklin Waddell. 2018. What does the crowd think? How online comments and popularity metrics affect news credibility and issue importance. *New Media & Society* 20, 8 (2018), 3068–3083.

[39] Heyuan Wang, Fangzhao Wu, Zheng Liu, and Xing Xie. 2020. Fine-grained interest matching for neural news recommendation. In *Proceedings of the 58th annual meeting of the association for computational linguistics*. 836–845.

[40] Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019. Neural news recommendation with multi-head self-attention. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 6389–6394.

[41] Chuhan Wu, Fangzhao Wu, Yongfeng Huang, and Xing Xie. 2022. Personalized News Recommendation: Methods and Challenges. *ACM Transactions on Information Systems (TOIS)* (2022).

[42] Chuhan Wu, Fangzhao Wu, Tao Qi, and Yongfeng Huang. 2021. Empowering news recommendation with pre-trained language models. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1652–1656.

[43] Fangzhao Wu, Ying Qiao, Jiun-Hung Chen, Chuhan Wu, Tao Qi, Jianxun Lian, Danyang Liu, Xing Xie, Jianfeng Gao, Winnie Wu, et al. 2020. Mind: A large-scale dataset for news recommendation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 3597–3606.

[44] Siqi Wu, Marian-Andrei Rizoiu, and Lexing Xie. 2018. Beyond views: Measuring and predicting engagement in online videos. In *Twelfth international AAAI conference on web and social media*.

[45] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)* 52, 1 (2019), 1–38.

[46] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.